Likelihood, numerical optimization and the Bootstrap

gean piere ventura cruz

2022-09-12

Task 1

Compute the score vector and the fisher information matrix function

a.

```
L <- function(theta, y, X, N){
  likelihood <- matrix(0, nrow = 1, ncol = N) #creating a vector 1xN
  for (i in 1:N){
    likelihood[i] <- (1/(1+exp(-X[i,]%*%theta)))**y[i]%*%(1-(1/(1+exp(-X[i,]%*%theta))))**(1-
y[i]) #using the formula p(xi)^yi*(1-p(xi)^1-yi)
  }
  return(likelihood)
}

#L(theta0,y,X,1000)

l <- function(theta, y, X, N){
  log_likelihood <- matrix(0, nrow = 1, ncol = N)
  for (i in 1:N) {
    log_likelihood[i] <- y[i]*log((1/(1+exp(-X[i,]%*%theta))))+(1-y[i])*(log(1-(1/(1+exp(-X
[i,]%*%theta))))) #taking the log of the func
  }
  return(log_likelihood)
}

#l(theta0,y,X,1000)

S <- function(theta, y, X, N){
  p <- matrix(0, nrow = 1, ncol = N)
  for (i in 1:N){
    p[i] <- 1/(1+exp(-X[i,]%*%theta)) #creating the p vector
  }
  transposed_p <- t(p) #transpose p vector
  score <- t(X)%*%(y-transposed_p)
  return(score)
}

#S(theta0,y,X,1000)

I <- function(theta, y, X, N){
  v <- matrix(0, nrow = 1, ncol = N)
  for (i in 1:N) {
    v[i] <- (1/(1+exp(-X[i,]%*%theta)))%*%(1-(1/(1+exp(-X[i,]%*%theta)))) #creating the v vec
tor and storing it in a diagonal matrix
  }
  D <- diag(as.vector(v))
  return(t(X)%*%D%*%X)
}

#I(theta0,y,X,1000)
```

b.

Compute the newton-raphson method in order to obtain the ML estimator

```
NR <- function(theta0, niter, y, X){
  old_theta <- theta0
  for (i in 1:niter){
    theta_ml <- old_theta+solve(I(old_theta,y ,X, 1000))%*%S(old_theta, y, X, 1000)
    old_theta <- theta_ml
  }
  return(theta_ml)
}
```

Task 2

Using the nr function we can now estimate the parameters that are provided by the glm with two digits of accuracy

```
modell <- glm(Resultat ~ Alder + Kon + Utbildare,
              data = data_individ,
              family = "binomial")

summary(modell)
```

```
##
## Call:
## glm(formula = Resultat ~ Alder + Kon + Utbildare, family = "binomial",
##     data = data_individ)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5786  -0.9712  -0.7555   1.2398   2.0089
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.336619   0.254014   1.325    0.185
## Alder                -0.045140   0.009185  -4.914 8.91e-07 ***
## KonMan                0.330364   0.135873   2.431    0.015 *
## UtbildareTrafikskola  1.052320   0.157491   6.682 2.36e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1355.9  on 999  degrees of freedom
## Residual deviance: 1271.5  on 996  degrees of freedom
## AIC: 1279.5
##
## Number of Fisher Scoring iterations: 4
```

```
y <- matrix(data_individ$Resultat, ncol = 1)
X <- model.matrix(Resultat ~ Alder + Kon + Utbildare, data = data_individ)



#head(data_individ[,-1])
theta0 <- c(0,0,0,0)
theta_est <- NR(theta0, niter = 3, y, X)
theta_est
```

```
##                             [,1]
## (Intercept)           0.33661607
## Alder                -0.04513943
## KonMan                0.33036377
## UtbildareTrafikskola  1.05231937
```

We seem to get the same approximation after two iterations with two digits of accuracy.

Task 3

in this part we will estimate the stde by using the fisher information matrix

```
ml_theta <- NR(theta0, niter = 3, y, X)
#solve(I(ml_theta,y,X,1000))
stde_test <-c()
stde_test[1] = solve(I(ml_theta,y,X,1000))[1,1]
stde_test[2] = solve(I(ml_theta,y,X,1000))[2,2]
stde_test[3] = solve(I(ml_theta,y,X,1000))[3,3]
stde_test[4] = solve(I(ml_theta,y,X,1000))[4,4]
sqrt(stde_test)
```

```
## [1] 0.254014333 0.009185207 0.135873174 0.157491273
```

Looking at the result from the stde and the std.error from the table we see that these result are similar so it seems like r is using the same method.

Task 4

we will estimate the stde but now with help of bootstraping. We are also going to construct a Bootstrap 95% confidence interval for the probability that someone privately educated of your own age and sex is successful.

```
set.seed(960618)
age <- c(round(runif(1000,18,49),0)) #1000 sumulations for ages between 18-49
utbildning <- c(rbinom(1000, 1, prob = 0.5)) #1000 simulations for utb either 1 or 0
kon <- c(rbinom(1000,1, prob = 0.5)) #the same here either 1 or 0
intercept <- c(rep(1,1000))

simul_sample <- matrix(0,nrow = 1000,ncol= 4) #creating a matrix 1000 x 4 for 1000 people
for (i in 1:1000) {
  simul_sample[i,1] <-intercept[i]
  simul_sample[i,2] <- age[i]
  simul_sample[i,3] <- kon[i]
  simul_sample[i,4] <- utbildning[i]
}

new_p <- matrix(0,nrow = 1, ncol = 1000) #creating a matrix for 1000 probabilities for the 10
00 people
for (i in 1:1000) {
  new_p[i] <- 1/(1+exp(-simul_sample[i,]%*%theta_est))
}
transposed_new_p <- t(new_p)

new_y <- replicate(1000,rbinom(1000,1,new_p))


new_theta_est <- matrix(0,nrow = 1000, ncol = 4)
for (i in 1:1000) {
  new_theta_est[i,] <- NR(theta0, niter = 3, new_y[,i],simul_sample)
}

#Our standard errors using bootstraping method:

sd_theta_1 <- sd(new_theta_est[,1])
sd_theta_2 <- sd(new_theta_est[,2])
sd_theta_3 <- sd(new_theta_est[,3])
sd_theta_4 <- sd(new_theta_est[,4])
sd_theta_1
```

```
## [1] 0.2796257
```

```
sd_theta_2
```

```
## [1] 0.007827463
```

```
sd_theta_3
```

```
## [1] 0.1355492
```

```
sd_theta_4
```

```
## [1] 0.1370617
```

We can see that the standard errors from task 3 are quite similar to the ones obtained via bootstrapping.

```
quantile(new_theta_est[,4], probs = c(0.025,0.975)) #getting the 2.5% and 97.5% quantiles
```

```
##      2.5%     97.5%
## 0.8019581 1.3356171
```

```
p_25 <- 1/(1+exp(-c(1,26,1,1)%*%c(theta_est[1],theta_est[2],theta_est[3],0.8019581)))
p_25
```

```
##           [,1]
## [1,] 0.5732963
```

```
p_975 <- 1/(1+exp(-c(1,26,1,1)%*%c(theta_est[1],theta_est[2],theta_est[3],1.3356171)))
p_975
```

```
##           [,1]
## [1,] 0.6961375
```

The probability that someone privately educated of my age and sex is successful with the probabilities 57% depending of the 2.5% quantiles and 69% of the 97.5% quantiles.